Your notes

GCSE Computer Science: revision notes

Subroutines and parameters

Contents

1. What subroutines are
2. Parameters and return values
3. Local and global variables
4. Why use subroutines?
5. Key terms
6. Quick questions
7. What to do next

What subroutines are

- A **subroutine** is a named block of code that carries out a specific task. It can be called (run) whenever it is needed.
- A **procedure** carries out a task. A **function** carries out a task and returns a value. (AQA calls both '**subroutines**'; Edexcel calls them 'subprograms'.)

Parameters and return values

- **Parameters** are the variables in the **subroutine**'s definition that receive data. The actual values passed in when it is called are called **arguments**.
- A **function** area(length, width) that returns length × width gives 12 when called as area(3, 4).

Local and global variables

- A **local variable** only exists inside its **subroutine**. A **global variable** can be used anywhere in the program.
- **Local variables** are good practice: the **subroutine** is self-contained, and its variable names can't clash with the rest of the program.

Why use subroutines?

- Code is written once and reused, so programs are shorter. Each **subroutine** can be tested separately, which makes debugging easier.
- Programs are easier to read and maintain, and different programmers can work on different **subroutines** at the same time.
- A structured approach uses **subroutines** with clear **parameters** and **return values**, and good documentation.

© LycoLearn · lycolearn.co.uk · Free GCSE and KS3 revision

Key terms

Subroutine	A named block of code that carries out a specific task.
Procedure	A subroutine that carries out a task.
Function	A subroutine that returns a value.
Parameter	A variable in a subroutine's definition that receives a value.
Argument	The actual value passed into a subroutine when it is called.
Return value	The value a function sends back to where it was called.
Local variable	A variable that only exists inside its subroutine.
Global variable	A variable that can be used anywhere in the program.

Quick questions

Cover the answers and test yourself: remembering is what makes it stick.

1. What is the difference between a function and a procedure?
2. What is a parameter?
3. A function area(length, width) returns length × width. What does area(3, 4) return?
4. What is a local variable?
5. Why is using local variables good practice?
6. Which of these is a benefit of using subroutines?
7. What is the value passed into a subroutine when it is called?
8. Which keyword sends a value back from a function?

Answers

1. A function returns a value; a procedure doesn't have to
2. A variable that receives a value passed into a subroutine
3. 12
4. A variable that only exists inside its subroutine
5. The subroutine is self-contained and names can't clash
6. Code can be reused instead of repeated
7. An argument
8. RETURN

What to do next

Practise 19 questions on this topic

Try 6 exam questions, with mark schemes

Read the lesson, one step at a time

Free revision for GCSE and KS3 at lycolearn.co.uk. Some notes are for one exam board only, and say so in brackets.

Your notes